Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Structured Design of Reconfigurable Logic Control Functions through Sequential Functional Charts

E. Carpanzano, A. Cataldo, and D. Tilbury, *Member, IEEE*

Abstract—**In the present work a modular logic control design methodology for Reconfigurable Manufacturing Systems is presented. The presented method guarantees reachability, liveness, and reversibility for the designed control solution. In particular, the logic control functions are designed through the Sequential Functional Charts graphical language, which is one of the programming languages included in the IEC 61131 standard for industrial Programmable Logic Controllers. The different steps of the proposed framework are presented with reference to a real size industrial application example, where the design of the logic control for an innovative shoe Reconfigurable Manufacturing System assembly line is discussed.**

I. INTRODUCTION

NOWADAYS industrial companies have to face frequent and unpredictable market changes. So, to stay competitive, companies must possess new types of production systems that are very responsive to all such changes, i.e. Reconfigurable Manufacturing Systems (RMS) [1]. In particular, reliable and flexible control systems are a crucial element to properly cope with the considered problem. Moreover, easy reuse and reconfiguration of already developed control solutions is also a fundamental element to reduce the cost and time needed to design and realize a new production system or to modify an existent one [2]. In such a context, the definition of development methodologies that support the structured design and verification of the whole control and supervision system of a RMS is mandatory [3], [4].

In the proposed approach a standard programming language for Programmable Logic Controllers (PLCs) – the Sequential Functional Chart (SFC) [8], [9] – is adopted for the modular and structured development of reconfigurable logic control functions. In detail, the SFC formalism is first introduced (section II). Then, the proposed design methodology is illustrated (section III) and its application to an innovative shoe manufacturing plant is discussed (section IV). Finally, some concluding remarks about open problems and future work are drawn (section V).

II. THE SFC FORMALISM

The Sequential Functional Chart (SFC) formalism, formerly also called Grafcet, was defined starting from Petri nets, so as to define a new graphical discrete event model suitable to describe logic control systems. Thus, there are many similarities in the structures and in the evolution rules of Petri nets (PNs) and SFCs [10]. The SFC formalism has also been included in the IEC 61131 part 3 standard in 1993 as one of the five standard programming languages for PLCs [11].

A SFC can be formally defined as a 4-tuple $SFC = (S, T, F, M_0)$, where:

$S = \{s_1, s_2, s_3 \cdots, s_m\}$ is a finite set of steps;

$T = \{t_1, t_2, t_3, \cdots, t_n\}$ is a finite set of transitions;

$F = (S \times T) \cup (T \times S)$ are sets of connections from steps to transitions and from transitions to steps;

$M_0 : S \rightarrow \{0,1,2,3,\cdots\}$ is the initial marking;

$S \cap T = \phi$ and $S \cup T \neq \phi$.

Because of the similarities between PNs and SFCs the Reachability, Liveness and Reversibility properties definitions for Petri nets can be directly applied also to the SFC formalism [1], [5], [6]. So, such properties can be defined with reference to a Sequential Function Chart as follows.

E. Carpanzano is with the Institute of Industrial Technologies and Automation, National Research Council, V.le Lombardia 20/A, 20131, Milan, Italy (phone: +39 02 23699914, fax: +39 02 23699915, e-mail: e.carpanzano@itia.cnr.it).

A. Cataldo is with the Institute of Industrial Technologies and Automation, National Research Council, V.le Lombardia 20/A, 20131, Milan, Italy (e-mail: a.cataldo@itia.cnr.it).

D. Tilbury is with the Mechanical Engineering Department, University of Michigan, Ann Arbor, USA, MI 48109-212 (e-mail: tilbury@umich.edu).

0-7803-8335-4/04/$17.00 ©2004 AACC

Reachability. A marking M_n is said to be reachable from a marking M_o if there exists a sequence of firings that transforms M_o to M_n. Reachability is a behavioral property since it depends upon the initial marking. An important issue in designing a manufacturing control system is whether the system can reach a specific state or not, therefore the reachability property is of interest in the study of such systems.

Liveness. A SFC is said to be *live* (or equivalently M_o is said to be a *live marking* for the $SFC = (S, T, F, M_0)$) if, no matter what marking has been reached from M_o, it is possible to ultimately fire *any* transition of the SFC by progressing through some further firing sequence. This means that a live SFC guarantees deadlock-free operation, no matter what firing sequence is chosen. This property guarantees that all the transitions are firable, and all the states of the logic control system which are modeled by the SFC steps can occur.

Reversibility. A SFC is said to be *reversible* if, for each marking M in $R(M_0)$, M_o is reachable from M. Thus, in a reversible SFC it is always possible to go back to the initial marking or state. In many applications, it is not necessary to go back to the initial state as long as it is possible to reach some proper (home) state. A marking M' is said to be a *home state* if, for each marking M in $R(M_0)$, M' is reachable from M. Reversibility implies that the system will have a cyclic behavior and will perform its functions repeatedly. It also characterizes the recoverability of the initial state from any state of the system.

Furthermore, as for the *Safeness* property, which means that in a PN the number of tokens in each place does never exceed one, this can assumed to be always true for an SFC, since in the SFC notation each step can only be active (contains one token) or not active (contains no token), that is a Boolean state (true/false, on/off, 1/0) is associated to each step. So, if a step in the SFC notation is associated to a place in the Petri Net formalism and an active step to a place containing one token, then the maximum number of tokens in every place of the PN associated to an SFC can not exceed one. This means that every SFC can be assumed to be safe.

III. THE PROPOSED MODULAR DESIGN METHODOLOGY

To design RMS logic control systems a modular and distributed approach is here adopted. Modularity allows reducing the complexity of the overall design problem by splitting it into simpler sub-problems which can be dealt with separately [2], [7], [9]. The different control modules are hierarchically distributed over the different software and hardware modules of the RMS control system. Such an approach simplifies the logic control design and enhances the reuse and reconfigurability of the developed control solutions [3], [12].

According to the proposed approach the logic control code is structured on three hierarchical levels: system, cell and machine. In particular, the system control level is composed by the aggregation of more cell control modules, according to the system lay-out. A cell control module is constituted by the aggregation of more machine control modules. Specifically, cells are composed by both operating machines and machines dedicated to parts transportation and handling, like conveyors, rotating tables and manipulators (Figure 1).

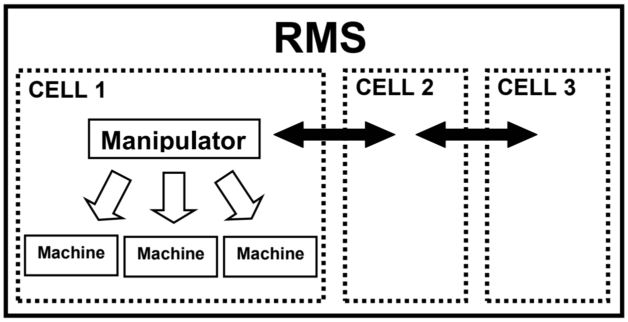


Fig. 1. Scheme of the modular and distributed control of a RMS.

The proposed RMS logic control development methodology follows a modular approach exploiting hierarchical SFC structures. In particular, complex SFC programs are developed so that the reachability, liveness and reversibility properties hold. Such programs are obtained by properly aggregating simpler SFCs for which the desired properties are satisfied. The considered design method is derived from similar known techniques based on the Petri net formalism, by exploiting the similarities between PNS and SFCs. Specifically, first the SFC for the logic control of every machine is designed through a top-down approach. Therefore the machine logic control program is initially defined by means of a simplified SFC characterized by the reachability, liveness and reversibility properties. Such an SFC is then refined further by refining its transitions and steps so as to obtain a more complex and more accurate SFC for which all the above mentioned properties hold. Then, the SFCs representing the cells control systems are obtained by properly aggregating the SFCs of the machines of the cells through a hierarchical structure. Finally, the whole RMS logic control SFC is defined by aggregating the SFCs of all the cells.

Here after the steps of the defined development methodology are outlined.

STEP 1: MACHINES CONTROL MODULES

Machine control modules are designed starting from simple SFC programs that implement the high level logic control of a machine and that are characterized by the reachability, liveness and reversibility properties. Such SFC programs are then detailed further according to the following rules.

- *Rule A.* A transition can be substituted by an SFC constituted by only one initial transition T_{ini} and only one final transition T_{fin}. In this case, if the new SFC is characterized by the reachability, liveness and reversibility properties then the refined SFC is still reachable, live and reversible.
- *Rule B.* A step can be substituted with a sequence: step-transition–step, and the new transition can be refined as pointed out in rule A. So maintaining the properties of interest for the overall SFC

STEP 2: CELLS CONTROL MODULES

The control modules of the RMS cells are obtained by following the design rules presented in step 1, and by associating to the cells SFC actions the proper execution of the SFCs representing the machines control modules. In particular, each machine control module has to be considered as a single resource that may be used (i.e. active) or not used (i.e. not active).

STEP 3: OVERALL SYSTEM CONTROL

The overall RMS control system is obtained by following the design rules presented in step 1, and by associating to the relative SFC actions the proper execution of the SFCs representing the cells control modules. Obviously, each cell control module has to be considered as a single resource that may be used (i.e. active) or not used (i.e. not active).

IV. APPLICATION TO AN INDUSTRIAL RMS

As an application example an innovative shoe production plant characterized by high-levels of flexibility and reconfigurability is here considered. The purpose of the plant is to produce customized shoes of different models in relation to the demand, reducing the time-to-market and the costs, and trying to maintain high-levels quality of the product. In figures 2 and 3 a picture of the plant and its lay-out are represented [2], [8], [12]. Such a plant is suited within the ITIA-CNR Laboratory of Vigevano.

In particular, the considered plant is characterized by an innovative transport line based on a molecular architecture and composed by six transport cells, called terns, since each cell is constituted by three devices: a table, an island and a manipulator (figure 4). The Terns move the parts to be worked while the workmanships are performed by the machines collocated around the Islands. An initial storehouse inserts the forms in the system, and a final storehouse picks up the produced shoes. The logic control of the innovative transport line has been designed by means of the proposed methodology. In particular, the modularity of the molecular architecture has been fully exploited by designing the logic control for a single tern, so that the tern itself acts as an autonomous agent by reacting to its boundary conditions, i.e. to the parts to be moved and/or worked, according to the control system specification. As a consequence, the same logic control scheme can be adopted for all terns, thus strongly enhancing reuse and reducing development times and efforts.

Fig. 2. Shoe manufacturing plant.

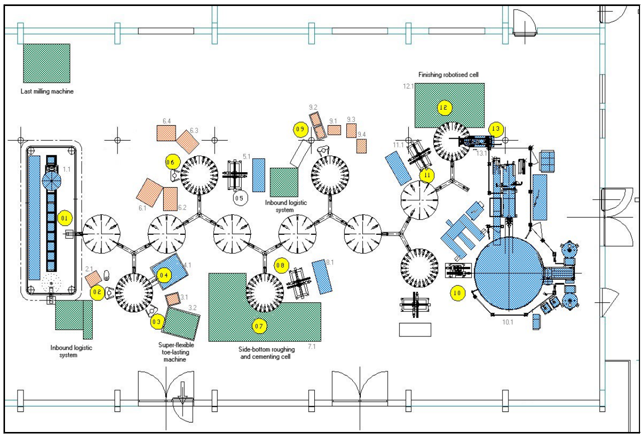


Fig. 3. Layout of the shoe manufacturing plant.

Now the design of the logic control for a single tern of the assembly line by means of the proposed methodology is discussed, going through the different design phases and highlighting the benefits of the proposed method as for reconfigurability of the logic control of the manufacturing system. In particular, the control solution has been implemented in the ISaGRAF environment. Such a soft logic tool supports all the IEC 61131 part 3 standard PLC programming languages, and provides editing, simulation and 2D graphic animation facilities.

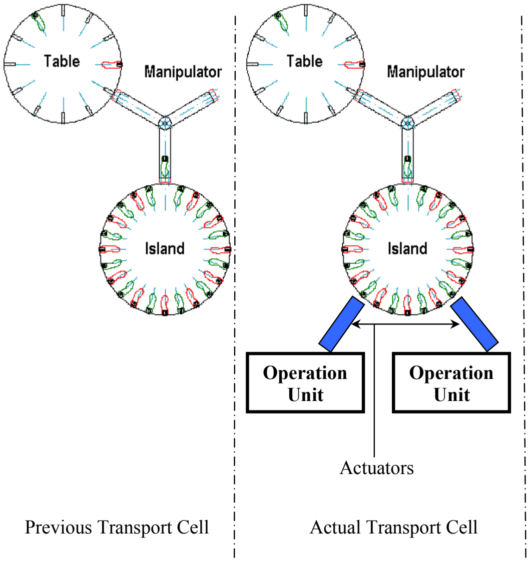


Fig. 4. Transport cells.

STEP 1: MACHINES CONTROL MODULES

First the SFCs of all the machines have been designed following the rules given in the previous section, so that the desired properties hold. In figure 5 the SFC for the manipulator is reported as an example.

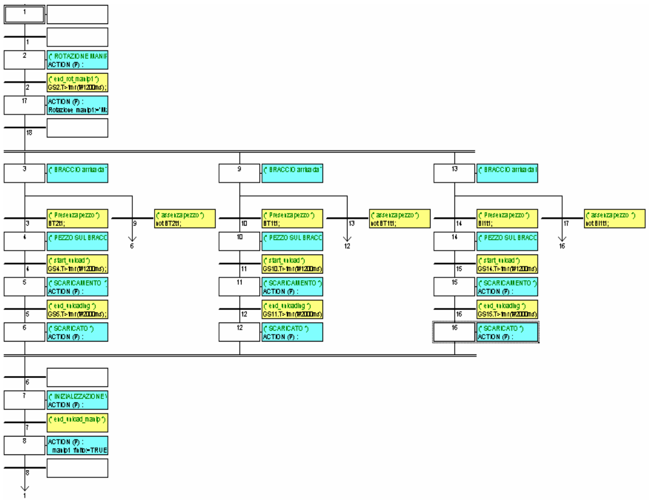


Fig. 5. SFC for the manipulator.

STEP 2: CELLS CONTROL MODULES

Once the machines control modules have been designed, and eventually tested, the cells control modules can be defined as outlined in section III. In this way a hierarchical program structure is defined as represented in figure 6, where the tern SFC is shown. So, the SFCs of the single machines constituting a cell are not directly and graphically connected by means of an SFC arc, but they are functionally connected by means of a higher level SFC program. In the definition of such a higher level SFC the rules A and B of the proposed methodology are exploited. Furthermore, resource sharing constraints are respected by properly developing the tern SFC, i.e. by defining proper sequences as for the activations of the machine control SFC modules and by avoiding multiple activations of the machine control SFC modules.

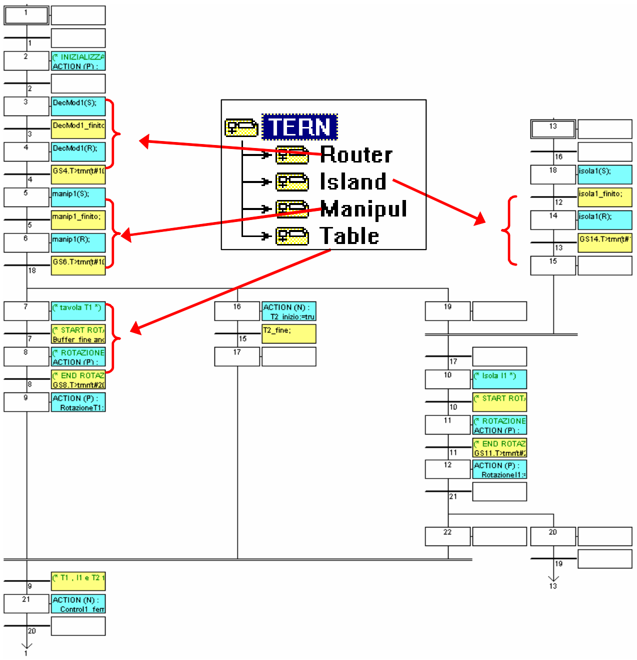
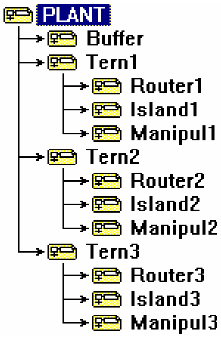


Fig. 6. SFC for the tern.

STEP 3: OVERALL SYSTEM CONTROL

Once the control modules for the different terns have been defined, they can be functionally connected by means of a higher level SFC program, according to the implementation mechanism discussed at the previous section. So, the overall molecular transport line SFC code can be obtained, and its structure, with reference to a system constituted by three terns, is depicted in figure 7.

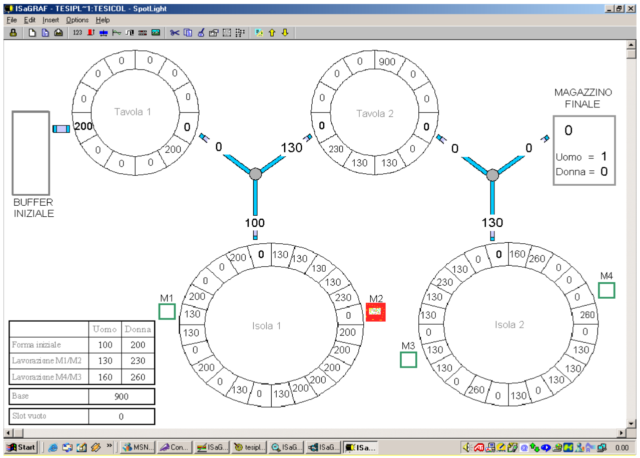


- PLANT
  - Buffer
  - Tern1
    - Router1
    - Island1
    - Manipul1
  - Tern2
    - Router2
    - Island2
    - Manipul2
  - Tern3
    - Router3
    - Island3
    - Manipul3

Fig. 7. Transport line control SFC program structure.

It may be noticed the proposed design method enhances reusability and reconfigurability of the defined control solution since it is possible to re-use, change or modify a single SFC control module while preserving the overall control solution as well as its functional properties.

In particular, the correctness and effectiveness of the defined control functions has been verified by means of closed loop discrete simulations. Therefore, also the controlled system discrete event behaviour has been represented by means of SFC models. Simulations have been performed by considering typical operating conditions, i.e. typical production orders, and the results obtained show that the system is deadlock free and that the plant is well balanced, i.e. its resources are all used effectively. In figure 8, the ISaGRAF graphic interface for the analysis of the simulation results is shown. Notice that for the sake of simplicity only two terns are represented.



	Uomo	Donna
Forma iniziale	100	200
Lavorazione M1/M2	130	230
Lavorazione M4/M3	160	260

Base	900
Slot vuoto	0

Fig. 8. Simulation and 2D animation in ISaGRAF.

V. CONCLUDING REMARKS

A structured and modular methodology is presented for the development of RMS logic control systems. Such a methodology adopts SFCs as a design language, in order to guarantee the properties of reachability, liveness, and reversibility hold for the designed control functions; and to implement the defined control solutions on industrial PLCs. An application example is also discussed throughout the paper. Future work will include the exploitation of the proposed methodology with reference to other manufacturing systems, and the definition and study of colored SFCs to extend the proposed methodology to more complex systems.

Acknowledgements. This research was supported in part by the NSF through the ERC/RMS grant number EEC95-92125.

REFERENCES

[1] M.C. Zhou and K. Venkatesh "Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach". New Jersey 1998 World Scientific.

[2] Carpanzano E. and A. Cataldo. "Modular design and simulation based verification of the logic control code for an agile shoe manufacturing system". Proc. Industrial Simulation Conference, ISC2003, EUROSIS, Valencia, Spain, June 9-11, 2003, pp. 290-294.

[3] Ferrarini, L. and Carpanzano, E. "A Structured Methodology for the Design and Implementation of Control and Supervision Systems for Robotic Applications". IEEE Transactions on Control Systems Technology, Vol. 10, N° 2, 2002, pp. 272-279.

[4] Carpanzano E. and Ballarino A. "A structured approach to the design and simulation-based testing of factory automation systems". Proc. IEEE International Symposium on Industrial Electronics, ISIE2002, L'Aquila, Italy, 8-11 July, 2002, pp. 181-186.

[5] Silva M. and R. Vanette, "Petri Nets and Flexible Manufacturing". In Advances in Petri Nets, Berlin, Springer Verlag, 1989, pp. 374-417.

[6] Murata T "Petri Nets: Properties, Analysis and Applications". Proc. Of the IEEE, vol. 77, n. 4, 1988.

[7] Park, E., Khargonekar, P., and Tilbury D. "A Modeling and Analysis Methodology for Modular Logic Controllers of Machining Systems Using Petri Nets Formalism". IEEE Transactions on Control Systems Technology, Vol. 31, N° 2, May 2001, pp. 168-186.

[8] Bossi D., Cattaneo della Volta V. "Modular Logic Control Design for Reconfigurable Manufacturing Systems through Petri nets and Sequential Function Charts". Master Thesis. Politecnico di Milano. April 2003.

[9] Park E. and Tilbury D. "Modular Logic Controllers for Machining Systems: Formal Representation and Performance Analysis using Petri Nets Formalism". IEEE Transactions on Control Systems Technology, Vol. 15, N° 6, December 1999, pp. 1046-1060.

[10] Renè David, "Grafcet: A Powerful Tool for Specification of Logic Controllers". IEEE Transactions on Control Systems Technology, Vol. N°3 September 1995.

[11] R.W. Lewis "Programming Industrial Control Systems using IEC 1131-3". Revised edition, The Institution of Electrical Engineers USA, 1998.

[12] Carpanzano E., Dallefrate D. and Jatta F. "A Modular Framework for the Development of Self-reconfiguring Manufacturing Control Systems". Proc. IEEE International Conference on Intelligent Robots and Systems, IROS2002, Lausanne, Switzerland, September 30-October 4, 2002, Vol. 2, pp. 1960-1965.